

**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL  
FACULDADE DE TECNOLOGIA SENAI/SC FLORIANÓPOLIS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**DEREK WILLIAN STAVIS**

**UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT TRANSMISSION  
BASEADO NO GTK HIG 3.14**

**Florianópolis/SC**

**2014**

**DEREK WILLIAN STAVIS**

**UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT TRANSMISSION  
BASEADO NO GTK HIG 3.14**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia do SENAI Florianópolis como requisito parcial para obtenção do Grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Professor Orientador: Aline Cristina Antoneli de Oliveira.

**Florianópolis/SC**

**2014**

**DEREK WILLIAN STAVIS**

**UMA PROPOSTA DE INTERFACE DO CLIENTE DE TORRENT TRANSMISSION  
BASEADO NO GTK HIG 3.14**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia do SENAI Florianópolis como requisito parcial para obtenção do Grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**APROVADA PELA COMISSÃO EXAMINADORA  
EM FLORIANÓPOLIS, 28 DE NOVEMBRO DE 2014**

---

Prof. Luciana Schmitz, Esp. (SENAI/SC)  
Coordenador do Curso

---

Profa. Jaqueline Voltolini de Almeida, Me. (SENAI/SC)  
Coordenador de TCC

---

Prof. Aline Cristina Antoneli de Oliveira, Dr. (SENAI/SC)  
Orientador

---

Prof. Fulado de tal, Me. (SENAI/SC)  
Examinador

## **AGRADECIMENTOS**

Em primeiro lugar, agradeço a meus pais. Sem eles eu não estaria aqui.

Em segundo, agradeço a internet, cada site e cada pessoa que pôde me fazer entender e conhecer algo que eu não sabia.

*“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível”*  
**(CHARLES CHAPLIN)**

STAVIS, Derek Willian. **Uma proposta de interface do cliente de torrent Transmission baseado no GTK HIG 3.14** Florianópolis, 2013. 26f. Trabalho de Conclusão de Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2013.

## RESUMO

**Palavras-chave:** Interfaces Gráficas. GTK+. GNOME. Design. Redesign. Os avanços trazidos nos últimos 4 anos no ambiente gráfico GNOME foram de suma importância para atingir uma experiência diferenciada e consistente de uso. Conceitos já difundidos no design de interfaces de usuário foram revisitados, paradigmas foram quebrados, mudando a forma com que os usuários vêem e interagem com o sistema operacional. Entretanto, nem todos os softwares disponíveis para a plataforma acompanharam a velocidade de desenvolvimento da plataforma GNOME, e muitos ainda necessitam de atenção. Com o lançamento do HIG (Guia de Interface com o Usuário) no GNOME versão 3.12 toda filosofia e padrões de design da plataforma foram sintetizados em linguagem simples e objetiva, tornando-se o material referência para o desenvolvimento e manutenção de seu ecossistema de softwares. Esta pesquisa aborda o processo de adaptação do cliente de torrent Transmission de acordo com o guia de interfaces do GNOME versão 3.14.

STAVIS, Derek Willian. **Uma proposta de interface do cliente de torrent Transmission baseado no GTK HIG 3.14** Florianópolis, 2013. 89f. Trabalho de Conclusão de Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2013.

## ABSTRACT

This is the english abstract.

**Key-words:** Graphical User Interfaces. GTK+. GNOME. Design. Redesign.

## **LISTA DE ILUSTRAÇÕES**

<b>Figura 1 – Uma captura de tela do GNOME 3.16 . . . . .</b>	<b>15</b>
<b>Figura 2 – Transmission 2.28 no GNOME 3.16 . . . . .</b>	<b>18</b>
<b>Figura 3 – Padrões e suas aplicações . . . . .</b>	<b>20</b>
<b>Figura 4 – Transmission no GNOME 3.16 . . . . .</b>	<b>22</b>

## **LISTA DE TABELAS**

## **LISTA DE ABREVIATURAS E SIGLAS**

GTK	GIMP Toolkit - Biblioteca de componentes para criação de interfaces gráficas.
GIMP	GNU Image Manipulation tool - Ferramenta de manipulações de imagem de código aberto, sob licença GNU.
GNOME	Ambiente gráfico de estação de trabalho disponível para GNU/Linux.
GNU/Linux	Distribuição de software de código aberto que forma a base de um sistema operacional.

## **LISTA DE SÍMBOLOS**

$\Gamma$  Letra grega Gama

$\Lambda$  Lambda

$\zeta$  Letra grega minúscula zeta

$\in$  Pertence

## SUMÁRIO

<b>1 INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1 OBJETIVOS . . . . .	12
<b>1.1.1 Objetivo geral . . . . .</b>	<b>12</b>
<b>1.1.2 Objetivos específicos . . . . .</b>	<b>12</b>
<b>2 REVISÃO DA LITERATURA . . . . .</b>	<b>13</b>
2.1 Interfaces gráficas . . . . .	13
2.2 Gerenciadores de Janelas . . . . .	14
2.3 Toolkits Gráficos . . . . .	14
2.4 A plataforma GNOME e o toolkit GTK . . . . .	15
2.5 Software livre e o desenvolvimento contínuo . . . . .	16
2.6 Impactos na consistência e usabilidade . . . . .	16
2.7 HIG ou Human Interface Guidelines . . . . .	17
2.8 Estudo de caso: Transmission . . . . .	17
<b>3 PROCEDIMENTOS METODOLÓGICOS . . . . .</b>	<b>19</b>
3.1 Análise cronológica dos padrões de design . . . . .	19
3.2 Configuração do Ambiente de Testes . . . . .	19
<b>4 RESULTADOS E DISCUSSÕES . . . . .</b>	<b>22</b>
4.1 A evolução das aplicações centrais analisadas . . . . .	22
4.2 Barra de Título . . . . .	22
4.3 Barra de Filtros . . . . .	23
4.4 Limites Globais de Upload/Download . . . . .	23
4.5 Botão Limite de velocidade . . . . .	23
4.6 Estísticas da barra de status . . . . .	23
<b>5 CONCLUSÃO . . . . .</b>	<b>24</b>
<b>REFERÊNCIAS . . . . .</b>	<b>25</b>

## **1 INTRODUÇÃO**

O lançamento de um guia de padrões para o desenvolvimento de interfaces gráficas orientadas ao ambiente gráfico GNOME foi de suma importância para alinhar desenvolvedores ao intuito deste ambiente gráfico de código aberto.

O guia contribui para que o GNOME passe a ter mais harmonia visual, seja através dos aplicativos centrais, – os primeiros a receber tratamento estético – ou por aplicativos de terceiros, que devem se adaptar utilizando o guia como referência.

Aplicativos que não se adequam ao guia passam a apresentar-se defasados, perdendo pontos em usabilidade e em beleza, e aumentando a taxa de rejeição. Mantenedores de aplicativos baseados no toolkit GTK devem, portanto, dedicar atenção para o fator de usabilidade e o progresso da plataforma quais foram desenhados para.

Um aplicativo que sofre deste efeito e necessita dedicação é o Transmission, um cliente de torrent popular, que foi desenvolvido na era GTK 2, onde os requisitos, recursos e a visão da plataforma GNOME eram diferentes (BENSON; CLARK; NICKELL, 2014).

O objetivo principal desta pesquisa é explorar o HIG do GTK 3.14, além de avançar o desenvolvimento de um software livre mantido pela comunidade de código aberto, propondo e contribuindo melhorias visuais e de código-fonte na interface gráfica do Transmission.

Esta pesquisa foi baseada na análise de diversos casos de uso de softwares contidos na plataforma GNOME Desktop, versão 3.14, e tem a pretensão de descrever o processo de pensamento e a motivação por trás das adaptações de interface gráficas propostas.

### **1.1 OBJETIVOS**

#### **1.1.1 Objetivo geral**

Propor adaptações na interface gráfica GTK do cliente de torrent Transmission de acordo com as recomendações do HIG versão 3.14, trazendo mais harmonia para seus utilizadores na plataforma GNOME.

#### **1.1.2 Objetivos específicos**

1. Comparar dados evolutivos da interface de aplicações centrais do GNOME.
2. Identificar pontos de redesign na janela principal do Transmission.
3. Propor adaptações na interface do Transmission, embasadas no HIG 3.14.

## 2 REVISÃO DA LITERATURA

O desenvolvimento de mecanismos que automatizam tarefas existe desde a idade das pedras, onde hominídeos produziam ferramentas para auxílio próprio. A transversalidade do conhecimento e da experimentação nos levou a descoberta de novas metodologias e o aperfeiçoamento de técnicas, consequentemente modificando a forma com que o ser humano interage com o mundo (SARTORI, 2010, p. 1).

### 2.1 Interfaces gráficas

O conceito de interface é extremamente amplo, e foi largamente difundido com o início dos estudos de interação humano-máquina. O avanço da industrialização permitiu com que máquinas extremamente complexas substituíssem seres humanos nas tarefas mais difíceis, deixando estes, seus operadores, apenas com a responsabilidade de pilotá-las de forma simples e segura.

O advento de tecnologias multimídias interativas como computadores, celulares e tablets elevou o patamar da criação de interfaces e gestão de tarefas ao estado da arte, agregando conhecimentos transversais de artistas visuais a músicos.

Ferreira, Barr e Noble (2005, p. 1) afirmam que a criação de “interfaces com o usuário ainda está mais para arte do que para ciência.”, e justificam, explicando que “A maior parte do design ou redesign é baseado em estudos empíricos ou protótipos, e ainda há muito pouco compreensão teórica ou de engenharia de como conduzir o processo de design e produzir bons designs pela primeira vez”.

Interfaces gráficas modernas foram alcançadas através da associação entre hardware e software, podendo utilizar de um ou mais dispositivos de entrada e saída, com o intuito de promover usabilidade e fácil adaptação.

Dispositivos de entrada transformam coordenadas do mundo real para o mundo virtual, registrando condições externas que podem ser modificadas através da interação com um ou mais atores. Dispositivos de entrada comuns são mouse, tela de toque, teclado, etc.

Dispositivos de saída projetam informações geradas por um sistema computacional e seu caráter é geralmente baseado nos sentidos: Visão, audição, tato. O dispositivo de saída mais comum é o monitor, que tem por finalidade projetar imagens compatíveis com a capacidade humana de visão.

## 2.2 Gerenciadores de Janelas

Gerenciadores de janelas tem como principal função dividir a imagem de um dispositivo de saída (geralmente monitores) em múltiplas regiões de desenho, popularmente denominadas janelas (MYERS, 1996, p. 5).

O primeiro conceito científico de gerenciamento de tarefas através da sobreposição de janelas data de 1969, na tese de Ph.D. de Alan Kay. A implementação de seu conceito, vista pela primeira vez em funcionamento no sistema do Xerox PARC, é largamente utilizada até hoje por grandes sistemas tanto comerciais quanto de código aberto, como Windows, OS X, GNOME, KDE (MYERS; HUDSON; PAUSCH, 2000, p. 7).

Um gerenciador de janelas moderno também tem por finalidade coordenar a exibição de um conjunto de janelas em um conjunto de monitores, escutar por eventos de entrada (mouse, teclado) e informar os responsáveis pelas janelas sobre alterações no layout de tela (dimensões da tela, dimensões da janela, espaço de cor).

Gerenciadores de janelas, porém, não tem por responsabilidade preencher as áreas de desenho com gráficos, e como suas APIs operam geralmente a nível de pixel, a tarefa de escrever um programa gráfico acaba sendo demorada e entediante. Além disso, se cada desenvolvedor criasse seus próprios componentes, seria praticamente impossível disponibilizar uma experiência consistente ao usuário Rosenthal (1988 apud MYERS; HUDSON; PAUSCH, 2000).

Para solucionar este problema ferramentas conhecidas como toolkits foram criadas sobre as abstrações disponibilizadas pelos WMs.

## 2.3 Toolkits Gráficos

Do ponto de vista do programador a finalidade de um toolkit é abstrair características de baixo nível, disponibilizando uma fachada homogênea e portável, além de comportamento e experiência visual consistente para o usuário final (MYERS; HUDSON; PAUSCH, 2000).

Do ponto de vista financeiro o reuso de código-fonte é uma maneira de amortizar o custo de desenvolvimento, reduzindo o tempo de desenvolvimento de novos projetos Haefliger, Krogh e Spaeth (2008).

As responsabilidades de um toolkit incluem desenhar elementos de interface gráfica como texto, botões, imagens, barras de progresso, etc, de acordo com um ou mais estilos visuais estes elementos de interface são chamados de widgets.

Também é sua responsabilidade processar eventos de um ou mais dispositivos de entrada (mouse, teclado, painel de toque) verificar a colisão de um evento com um widget (clique em um botão, por exemplo) e por fim traduzir e informar os eventos para a aplicação proprietária da janela.

Toolkits multiplataforma podem ser utilizados para escrever interfaces gráficas portáveis, permitindo com que o mesmo código seja recompilado para um sistema operacional diferente do em que foi escrito e funcione da mesma forma.

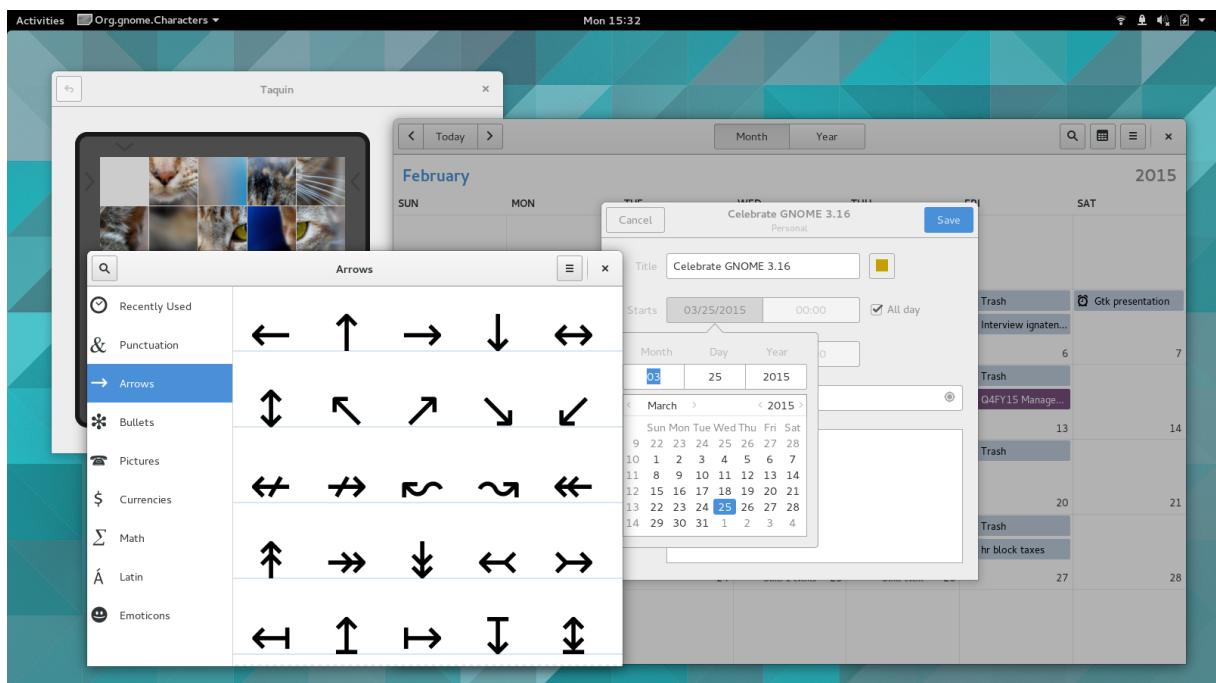
## 2.4 A plataforma GNOME e o toolkit GTK

Existem diversas opções de gerenciadores de janela de código aberto, comumente incluídos em várias distribuições de Linux. O GNOME, um ambiente gráfico bastante difundido pelos usuários de Linux, foi fundado e está em ativo desenvolvimento por uma comunidade de engenheiros de software ao redor do mundo.

O GNOME 3 é uma maneira fácil e elegante de usar seu computador. Ele foi desenhado para lhe colocar no controle e trazer liberdade para todos. O GNOME 3 é desenvolvido pela comunidade GNOME, um grupo internacional e diverso de contribuidores que são suportados por uma fundação independente, e sem fins lucrativos (THE GNOME PROJECT, 2015).

Muito mais que um gerenciador de janelas, a experiência GNOME é construída por um conjunto de aplicações centrais, que incluem um gerenciador de janelas, um lançador de aplicações e diversos aplicativos centrais, como calculadora, editor de texto, gerenciadores de arquivos, redes, contatos, etc.

**Figura 1 – Uma captura de tela do GNOME 3.16**



Fonte: Matias Clasen

A plataforma GNOME e seus aplicativos centrais são baseados no toolkit GTK. O GTK + é formado por um conjunto de ferramentas multi-plataforma para criar interfaces gráficas de

usuário. Por oferecer um conjunto completo de widgets é adequado para projetos desde pequenas ferramentas pontuais até suítes completas de aplicativos (THE GTK+ PROJECT, 2015).

## 2.5 Software livre e o desenvolvimento contínuo

Uma das característica das plataformas de código aberto é a distribuição de esforços em prol do constante desenvolvimento e melhoria. A pluralidade de opiniões e idéias eleva o patamar das discussões e permite com que vários pontos de vista sejam levados em consideração na evolução da plataforma.

Apesar dos prós existentes na distribuição de esforços também existem os contras – Projetos que são desenvolvidos paralelamente nem sempre avançam na mesma velocidade. O contra fica mais sério quando um projeto depende do outro, como é o caso de toolkits e programas que consomem suas APIs.

Quando ocorrem mudanças na interface de programação de uma framework ou biblioteca, softwares dependentes tem de se adaptar as mudanças. O efeito cascata provocado pela propagação de alterações na malha de softwares dependentes não é incomum, e já foi objeto de estudo (YAU; COLLOFELLO; MACGREGOR, 1978).

## 2.6 Impactos na consistência e usabilidade

Além dos aplicativos centrais, geralmente garantidos de acompanhar a evolução da plataforma – sua ergonomia e visual – existe uma vasta gama de aplicações tanto de código aberto quanto proprietárias disponíveis para atender as mais variadas necessidades.

A grande maioria das aplicações GTK+ são mantidas pela comunidade de software livre, e se não atualizadas podem ficar defasados em usabilidade, ergonomia e consistência com a plataforma.

Shneiderman (1992) descreve a usabilidade como uma combinação das seguintes características:

- Facilidade de aprendizado
- Alta velocidade de operação
- Baixa taxa de erros
- Satisfação do usuário
- Retenção de usuários pelo tempo

Em prol da usabilidade a maioria dos toolkits aplica um modelo próprio de apresentação, organização, interação e estilização de widgets, que costumeiramente varia de acordo com o tipo de ambiente onde é executado (desktop, tablet e celular).

## 2.7 HIG ou Human Interface Guidelines

Divulgado no ano de 2014, acompanhando a versão 3.14 da plataforma GNOME, sob o título de Human Interface Guidelines, o conjunto de padrões de design oficialmente promovido pela GNOME Foundation busca promover a máxima integração de interfaces gráficas GTK+ na plataforma GNOME.

Se você é um desenvolvedor com experiência limitada de design o HIG foi planejado para auxiliar você a criar facilmente uma interface de usuário efetiva. Para designers, o HIG provê uma introdução as possibilidades ao usar o GTK+, assim como padrões de design que são usados nos aplicativos GNOME (DAY, 2014a).

O HIG, como também é chamado, é uma literatura ilustrada de diretrizes recomendadas no desenvolvimento de interfaces gráficas que utilizem o toolkit, com o intuito de reforçar a consistência visual e integração com diferentes gerenciadores de janela.

Dentre os objetivos do HIG estão a transmissão das metas de design de alto nível e estratégicas da experiência GNOME e a comunicação das diretrizes de design essenciais, de uma forma clara, e viva, acompanhando a evolução da plataforma ??).

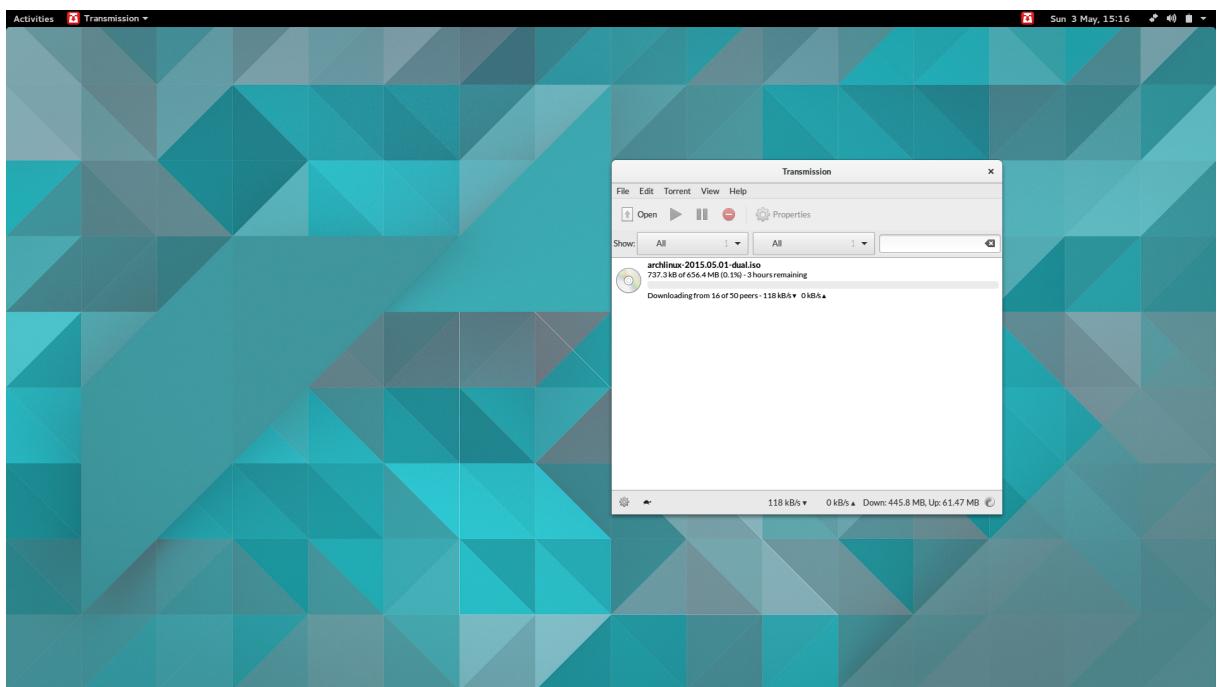
## 2.8 Estudo de caso: Transmission

Um aplicativo famoso disponível para a plataforma GNOME é o cliente de BitTorrent Transmission (The Transmission Project, ). Aclamado pela sua simplicidade, o aplicativo é utilizado para compartilhar arquivos através da internet.

Através de uma configuração padrão funcional e apenas alguns cliques para configurar funcionalidades avançadas a utilização do cliente de torrent Transmission foi projetada para ser fácil e poderosa (THE TRANSMISSION PROJECT, 2015).

A primeira versão do Transmission foi lançada em Setembro de 2005, já com uma interface gráfica baseada em GTK+, projetada sob as recomendação do HIG da época, publicada digitalmente em formato de livro (BENSON; CLARK; NICKELL, 2014).

**Figura 2 – Transmission 2.28 no GNOME 3.16**



Fonte: Transmission versão 2.28

### 3 PROCEDIMENTOS METODOLÓGICOS

Em um primeiro instante foram levantadas aplicações centrais do GNOME 3.16 de conhecimento popular e que já implementam os padrões de design definidos no HIG 3.14. Pelo importante papel desempenhado na experiência GNOME, foram selecionadas três aplicações centrais:

- Nautilus
- Evince
- Gedit

#### 3.1 Análise cronológica dos padrões de design

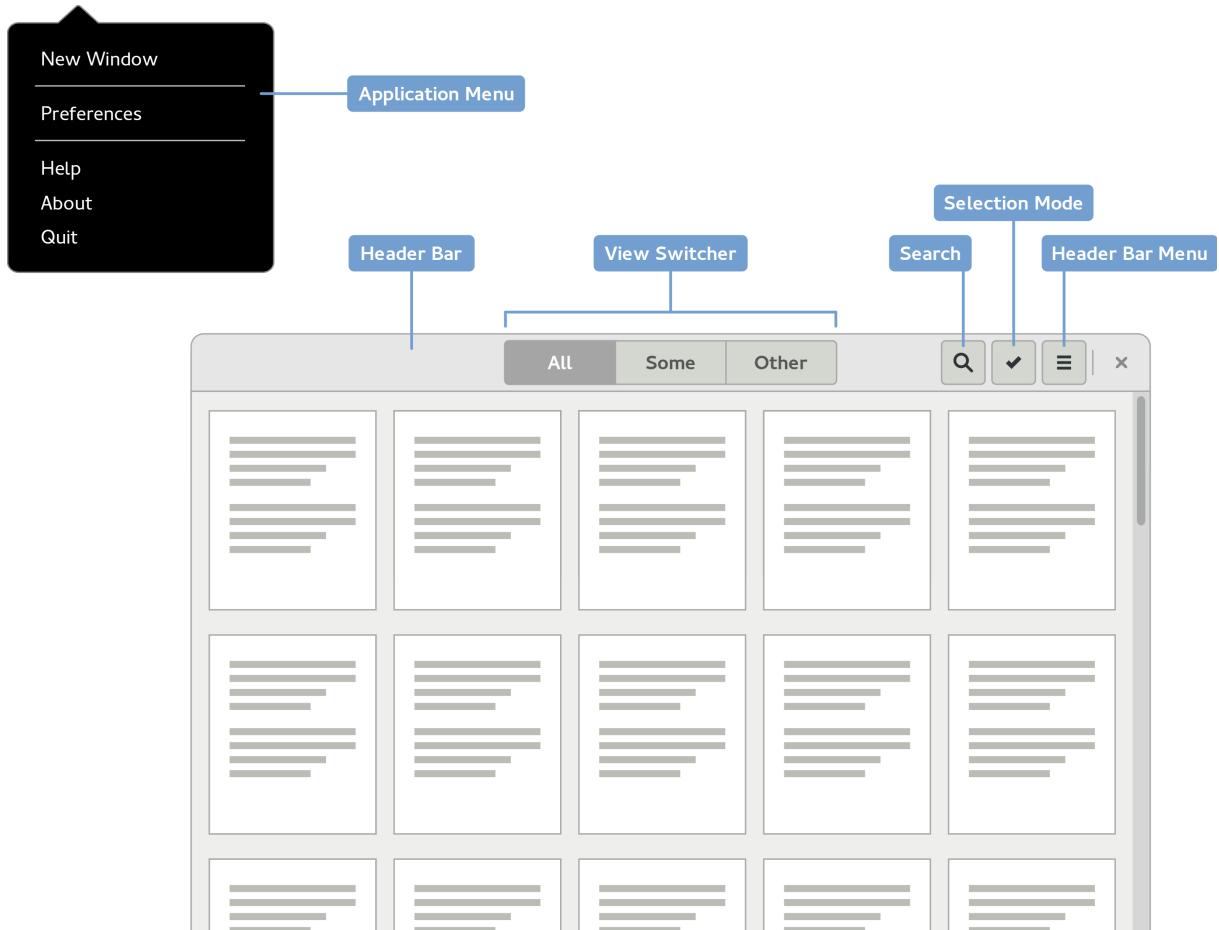
Dentre os padrões de design especificados pelo HIG (DAY, 2014b) foram escolhidos os seguintes a serem observados:

1. Menu da Aplicação
2. Janela Primária
3. Barra de Título
4. Comutador de Visão
5. Busca

#### 3.2 Configuração do Ambiente de Testes

O projeto GNOME disponibiliza imagens oficiais de um sistema operacional Linux com todos os pacotes necessários para uma experiência GNOME pré instalados e configurados. Sua utilização trouxe facilidade e consistência nos testes efetuados.

Dentre as versões disponíveis foram selecionadas :

**Figura 3 – Padrões e suas aplicações**

Fonte: GNOME 3.16 HIG - Patterns

Versão do GNOME	Data de Modificação	Nome do arquivo	Soma de Verificação
3.6.0	08/10/2012	GNOME-3.6.0.iso	MD5 753c99ce2342f658 65c1f74bc3722e44
3.16	25/03/2015	gnome-3.16.x86-64.iso	SHA256 4a6185a0aca89f15 8f769d76d5a0086f 0f1e9d709a5d80cd cf2b0d52d67ab2b2

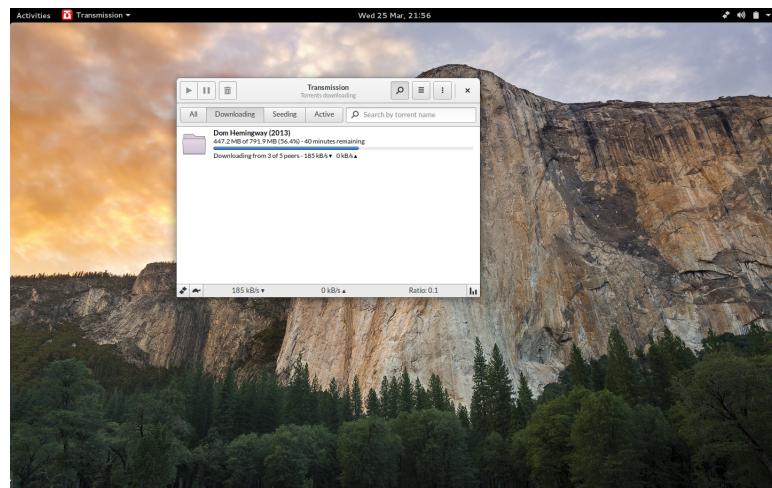
Todas aplicações foram analisadas na versão publicada, sem alterações na configuração do sistema, troca de temas, fontes, etc. O ambiente virtual foi provisionado utilizando o software Oracle VM VirtualBox 4.3.26 na seguinte configuração de máquina:

Máquina	
Memória Base	4096MB
Processadores	2
Aceleração	VT-x/AMD-V, Nested Paging, PAE/NX
Vídeo	
Memória	128MB
Aceleração	3D
Sistema Operacional	
Tipo	Linux
Versão	Fedora

## 4 RESULTADOS E DISCUSSÕES

O cenário ilustrado na 4 evidencia que a adaptação proposta na interface do Transmission provê maior harmonia com outros aplicativos da plataforma GNOME.

**Figura 4 – Transmission no GNOME 3.16**



### 4.1 A evolução das aplicações centrais analisadas

A análise das aplicações centrais do GNOME trouxe resultados efetivos sobre a transição gráfica e funcional dos principais widgets utilizados pela aplicação.

Um ítem notável foi a redução na utilização de espaço vertical em aplicações centrais. Day (2015) analisa a redução do uso vertical de espaço em várias versões do Nautilus e apresenta a seguinte tabela:

Versão do Nautilus	Moldura Vertical
2.22	171 pixels
3.0	116 pixels
3.6	70 pixels
3.12	48 pixels

Observou-se que o fenômeno se repete em todas as aplicações centrais analisadas:

### 4.2 Barra de Título

Um dos elementos de interface mais notáveis é a barra de título. Até então desenvolvedores tinham pouco ou nenhum controle sobre elas por questões de compatibilidade de software.

Com o surgimento do conceito de ‘Header Bars’, as barras de título ganharam botões, caixas de texto, sliders, etc.

#### 4.3 Barra de Filtros

Mostrar todos os controles de uma só vez torna a aplicação mais difícil de usar. Portanto é necessário

#### 4.4 Limites Globais de Upload/Download

#### 4.5 Botão Limite de velocidade

#### 4.6 Estísticas da barra de status

## 5 CONCLUSÃO

## REFERÊNCIAS

- BENSON, C.; CLARK, B.; NICKELL, S. *GNOME Human Interface Guidelines*. 2014. GNOME Human Interface Guidelines 2.2.1. Disponível em: <<https://developer.gnome.org/hig-book/2.32>>. Acesso em: 5 apr. 2015. Citado 2 vezes nas páginas 12 e 17.
- DAY, A. *GNOME Human Interface Guidelines*. 2014. GNOME Human Interface Guidelines. Disponível em: <<https://developer.gnome.org/hig/3.14>>. Acesso em: 5 apr. 2015. Citado na página 17.
- DAY, A. *Patterns*. 2014. GNOME Human Interface Guidelines. Disponível em: <<https://developer.gnome.org/hig/3.14>>. Acesso em: 5 apr. 2015. Citado na página 19.
- DAY, A. *GNOME design: saving space since 2009 (or so)*. 2015. GNOME design: saving space since 2009 (or so). Disponível em: <<https://blogs.gnome.org/aday/2014/08/27/gnome-design-saving-you-space-since-2009-or-so/>>. Acesso em: 3 may. 2015. Citado na página 22.
- FERREIRA, J.; BARR, P.; NOBLE, J. The semiotics of user interface redesign. In: AUSTRALIAN COMPUTER SOCIETY, INC. *Proceedings of the Sixth Australasian conference on User interface-Volume 40*. [S.I.], 2005. p. 47–53. Citado na página 13.
- HAEFLIGER, S.; KROGH, G. V.; SPAETH, S. Code reuse in open source software. *Management Science*, INFORMS, v. 54, n. 1, p. 180–193, 2008. Citado na página 14.
- MYERS, B.; HUDSON, S. E.; PAUSCH, R. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM, v. 7, n. 1, p. 3–28, 2000. Citado na página 14.
- MYERS, B. A. Uimss, toolkits, interface builders1. 1996. Citado na página 14.
- ROSENTHAL, D. A simple x11 client program-or-how hard can it really be to write "hello, world"? In: *USENIX Winter*. [S.I.: s.n.], 1988. p. 229–242. Citado na página 14.
- SARTORI, R. Neurociência e comportamento na educação de criança e adolescentes. *Cadernos de Pesquisa*, 2010. Citado na página 13.
- SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. [S.I.]: Addison-Wesley Reading, MA, 1992. Citado na página 16.
- THE GNOME PROJECT. *GNOME*. 2015. GNOME. Disponível em: <<https://gnome.org>>. Acesso em: 8 apr. 2015. Citado na página 15.
- THE GTK+ PROJECT. *The GTK+ Project*. 2015. The GTK+ Project. Disponível em: <<http://gtk.org>>. Acesso em: 10 apr. 2015. Citado na página 16.
- The Transmission Project. *Transmission*. Disponível em: <<http://transmissionbt.com>>. Citado na página 17.
- THE TRANSMISSION PROJECT. *About Transmission*. 2015. About. Disponível em: <<http://www.transmissionbt.com/about/>>. Acesso em: 3 apr. 2015. Citado na página 17.

YAU, S. S.; COLLOFELLO, J. S.; MACGREGOR, T. Ripple effect analysis of software maintenance. In: *Computer Software and Applications Conference, 1978. COMPSAC'78. The IEEE Computer Society's Second International*. [S.l.: s.n.], 1978. p. 60–65. Citado na página 16.